

# 32장 string

## String 생성자 함수

```
const strObj = new String();  
console.log(strObj); // String {length: 0, [[PrimitiveValue]]: ""}
```

## length 프로퍼티

length 프로퍼티는 문자열의 문자 개수를 반환

배열과 마찬가지로 length 프로퍼티를 갖는다

## String 메서드

### String.prototype.indexOf

대상 문자열(메서드를 호출한 문자열)에서 인수로 전달받은 문자열을 검색하여 첫 번째 인덱스 반환

```
const str = 'Hello World';  
  
// 문자열 str에서 'l'을 검색하여 첫 번째 인덱스를 반환한다.  
str.indexOf('l'); // 2  
  
// 문자열 str에서 'or'를 검색하여 첫 번째 인덱스를 반환한다.  
str.indexOf('or'); // 7
```

### String.prototype.search

대상 문자열에서 인수로 전달받은 정규 표현식과 매치하는 문자열을 검색, 문자열의 인덱스를 반환

```
const str = 'Hello world';
```

// 문자열 str에서 정규 표현식과 매치하는 문자열을 검색하여 일치하는 문자열의 인덱스를 반환한다.

```
str.search(/o/); // 4
```

### **string.prototype.includes(ES6)**

대상 문자열에 인수로 전달받은 문자열이 포함되어 있는지 확인하여 결과를 true 또는 false로 반환

```
const str = 'Hello world';
```

```
str.includes('Hello'); // true
```

```
str.includes(''); // true
```

### **String.prototype.substring**

대상 문자열에서 첫 번째 인수로 전달받은 인덱스에 위치하는 문자부터 두 번째 인수로 전달받은 인덱스에 위치하는 문자의 바로 이전 문자까지의 부분 문자열을 반환

```
const str = 'Hello World';
```

// 인덱스 1부터 인덱스 4 이전까지의 부분 문자열을 반환한다.

```
str.substring(1, 4); // ell
```

### **String.prototype.slice**

substring 메서드와 동일하게 동작

```
const str = "hello world";
```

// substring과 slice 메서드는 동일하게 동작한다.

// 0번째부터 5번째 이전 문자까지 잘라내어 반환

```
str.substring(0, 5); // -> 'hello'
```

```
str.slice(0, 5); // -> 'hello'
```

```
// 인수 < 0 또는 NaN인 경우 0으로 취급된다.  
str.substring(-5); // -> 'hello world'  
// slice 메서드는 음수인 인수를 전달할 수 있다. 뒤에서 5자리를 잘라내  
어 반환한다.  
str.slice(-5); // → 'world'
```

### **String.prototype.toUpperCase**

대상 문자열을 모두 소문자로 변경한 문자열을 반환

```
const str = 'Hello World!';  
  
str.toLowerCase(); // 'hello world!'
```

### **String.prototype.trim**

대상 문자열 앞뒤에 공백 문자가 있을 경우 이를 제거한 문자열을 반환

```
const str = '  foo  ';  
str.trim(); // 'foo'
```

### **String.prototype.replace**

replace 메서드는 대상 문자열에서 첫 번째 인수로 전달받은 문자열 또는 정규 표현식을 검색하여 두 번째 인수로 전달한 문자열로 치환한 문자열을 반환

```
const str = 'Hello world';  
  
// str에서 첫 번째 인수 'world'를 검색하여 두 번째 인수 'Lee'로 치환  
한다.  
str.replace('world', 'Lee'); // 'Hello Lee'
```

### **String.prototype.split**

대상 문자열에서 첫 번째 인수로 전달한 문자열 또는 정규 표현식을 검색하여 문자열을 구분한 후 분리된 각 문자열로 이루어진 배열을 반환

```
const str = 'How are you doing?';

// 공백으로 구분(단어로 구분)하여 배열로 반환한다.
str.split(' '); // ["How", "are", "you", "doing?"]

// \s는 여러 가지 공백 문자(스페이스, 탭 등)를 의미한다. 즉, [\t\r\n\v\f]와 같은 의미다.
str.split(/\s/); // ["How", "are", "you", "doing?"]
```